i - trOnics GmbH

Multi-channel Sampling Analogue-to-Digital Converter Module

i - SADC108032

for the Mainz Crystal Ball Detector

User's Manual

This manual was written to guide you in the installation, configuration and operation of the **i-SADC108032** multi-channel sampling analogue-to-digital converter module.

i-trOnics GmbH excludes completely any liability for any indirect, special, incidental, or consequential damages (including damages for loss of profits, loss of business, loss of use or data, interruption of business and the like) even if **i-trOnics GmbH** has been advised of the possibility of such damages arising from any defect or error in this manual or product.

Information and specifications in this manual are for informational use only. They are subject to change at any time without notice, and should not be construed as a declaration of conformity for this product nor as a commitment by **i-trOnics GmbH**. **i-trOnics GmbH** assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual, including the products and software described in it.

October 2003, Rev. 0.96

i-trOnics GmbH

Erikastrasse 18b
D-85521 Hohenbrunn
Germany

Tel:	+49-89-66006899	
Fax:	+49-89-66000377	
Internet:	`www.i-tronics.net`	
Email:	`info@i-tronics.net`	for general information
	`sales@i-tronics.net`	for sales information

Copyright ©2003 i-trOnics GmbH. All Rights Reserved

Product warranty or service will not be extended if: (i) the product is repaired, modified or altered, unless such repair, modification or alteration is authorised by **i-trOnics**; or (ii) the serial number of the product is defaced or missing.

Products and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are used only for identification or explanation and to the owner's benefit, without intent to infringe.

Contents

1 FCC statements and CE conformity declaration **1**

2 Safety information **2**

3 Specifications summary **3**

4 Scope of supply **4**

5 Hardware installation **5**

6 Overview **7**

6.1 The board 7

6.2 The front-panel and the analogue input 9

6.3 The VME backplane 10

6.4 Data processing and data flow 11

7 Data format **14**

7.1 Decoding the data 15

8 Configuration and loading **18**

8.1 Loading the HL FPGA 18

8.2 Loading the ZR FPGAs 19

8.3 Configuring the ZR FPGAs 19

1 FCC statements and CE conformity declaration

2 Safety information

Before starting with the installation of the **i-SADC108032** module, please verify that you received the shipment is complete and in perfect condition. Do not try to install modules that have visible defects, it could damage other components.

- Always use insulated, antistatic tools and avoid touching the electronic components.
- Before using the module in a crate, make sure all cables are connected correctly and the power cables are not damaged.
- To prevent electrical shock hazard, disconnect the power from the electrical outlet before relocating the system.
- When you add or remove modules from your system, ensure that the crate is switched off and that the power supply cable is out of the socket.
- Do not use force when plugging/unplugging modules.
- To avoid short circuits keep small conductive parts like screws, clips, and staples away from connectors, slots, sockets and circuitry.
- Avoid dust, humidity, and temperature extremes. Do not place the module in any area where it may become wet.

3 Specifications summary

Form-factor	VME 6U
Power inputs	+5 V/max. 5 A −12 V/max. 0.8 A
Power consumption	35 W total +5 V/25 W −12 V/10 W
Number of channels	32
ADC resolution	10 bit
Sampling rate	40 MHz
Noise performance	0.6 LSB RMS (DC input signal)
Input impedance	50 Ω
Input range	-1.25 V to 0.25 V
Input signal connector	2 × 34 pin
Maximum trigger latency	511 cycles = 12.78 μsec
Maximum number of samples/event	128
Depth of multi-hit buffer	4
Number of sample integrals	3
Data bandwidth of optical link	40 MByte/sec
Bandwidth for configuration data	40 MBit/sec
Maximum length of optical fibre	100 m

4 Scope of supply

Please check the contents of your package. The scope of supply of a **i-SADC108032** module includes:

- 1 × **i-SADC108032** multi-channel sampling ADC VME 6U board
- 1 × manual

If your package is not complete or is damaged, please contact **i-trOnics**.

5 Hardware installation

Attention:

- Make sure that you are using a standard VME crate with IEEE-1014 back plane.
- Make sure the VME crate has a fan unit which provides sufficient air cooling. Do not operate the module without active cooling.
- Make sure that unused slots of the VME crate are closed with metal screens to improve ventilation and electromagnetic emission.
- Always use insulated, antistatic tools and avoid touching the electronic components.
- Do not install modules with visible defects.
- Do not hot-plug modules.
- Do not use force to insert/remove modules.
- Avoid dust and dirt on the optical fibre connectors. Always use the protection caps for unconnected plugs and jacks.
- Carefully read the safety information in chapter 2.

Procedure:

1. Switch off the VME crate and pull the power supply cable out of the socket.
2. Carefully insert the **i - SADC108032** module in the VME slot and screw the front-panel to the crate.
3. Connect the cables for the analogue signals. Pin 1 sits in the lower right corner of the 34 pin connector (see subsection 6.2).
4. Reconnect the power supply cable and switch on the crate.
5. Make sure that the rightmost FPGA LED on the front-panel is lit (see subsection 6.2). If this is not the case the PROM for the HL FPGA has to be loaded (see subsection 8.1).

6. Setup and load the i-MUX module to which the i-SADC will be connected to.

7. Connect the fibre from the i-MUX module to the optical transceiver. The optical cable used must be crossed so that the transmitter (Tx) of the i-MUX transceiver is connected with the receiver (Rx) of the i-SADC module and vice versa.

8. Make sure that the 'Clk' LED on the front-panel (see subsection 6.2) is lit to indicate that the optical link was established successfully.

9. Check that the 'Syn' LED on the front-panel (see subsection 6.2) is active indicating that the clock recovery chip is active.

10. Load the ZR FPGAs (see subsection 8.2).

11. Set the configuration parameters of the ZR FPGAs (see subsection 8.3).

6 Overview

6.1 The board

The i-**SADC108032** module is made according to the 6U VME mechanical standard. A schematic is shown in figure 1.

Two 34 pin flat cable connectors bring the analogue differential signals to the 32 Analog Devices AD9214-80 analogue-to-digital converters.

The digital samples are read by two Xilinx XC2V1000 FPGAs named 'ZR0' and 'ZR1'. The ZR chips perform the digital data processing (latency compensation, buffering, zero suppression, data formatting) and each of them is connected to 16 channels: ZR0 to input channels 0 to 15, ZR1 to input channels 16 to 31.

The data go from the ZR chips over 16 bit wide links to the so called 'HL' chip. This Xilinx XCV100E FPGA multiplexes the data from the ZR FPGAs and writes them out to the HOTLink Transmitter which serialises and encodes the data so that they can be sent over the optical transceiver to the destination module. The HL chip also provides the interface to load and configure the two ZR FPGAs.

The IEEE-1014 back plane of the VME crate is only used to provide +5 V and −12 V power.

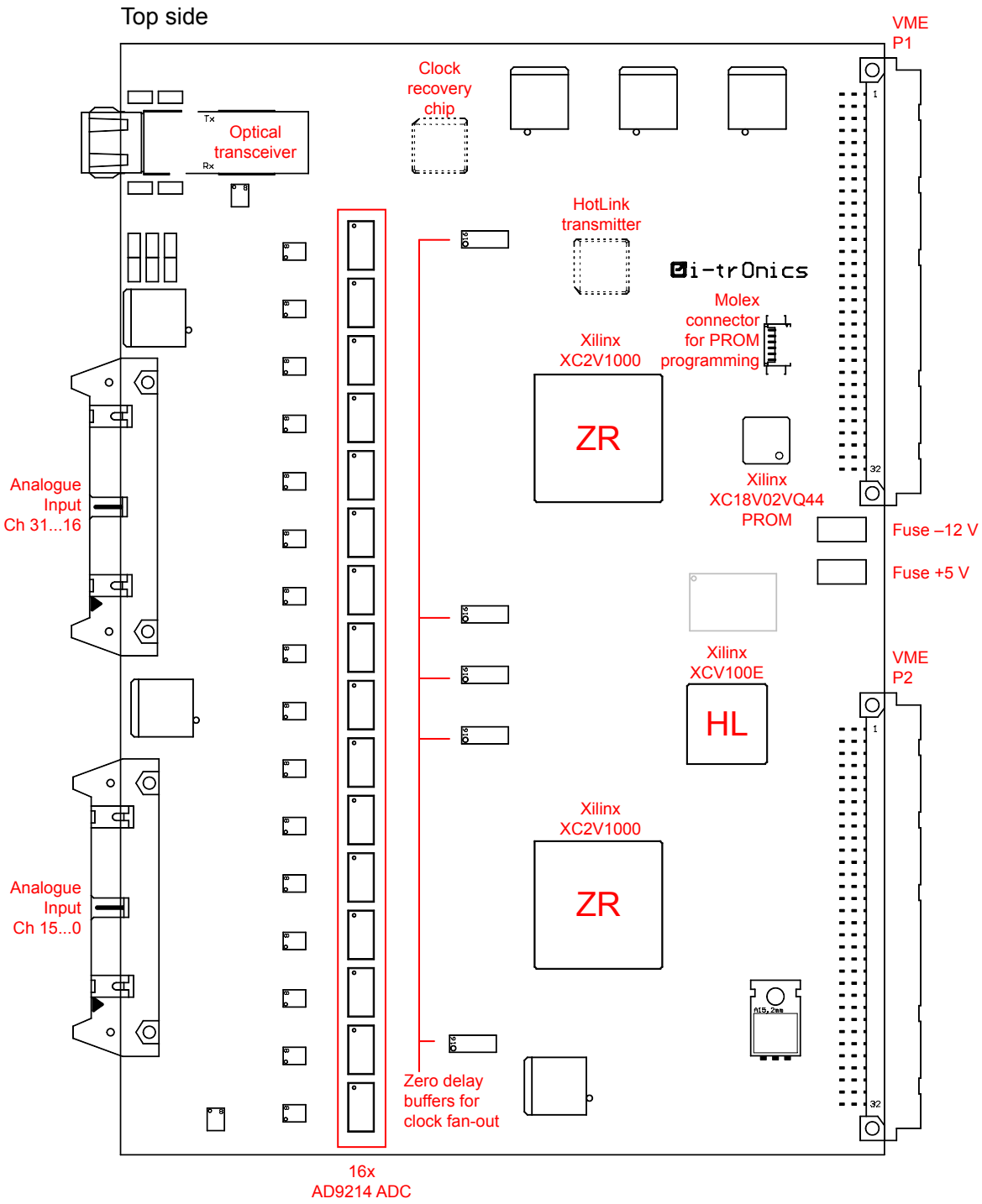

Figure 1: Board schematic

6.2 The front-panel and the analogue input

The front-panel of the **i-SADC108032** module is shown in figure 2 has three connectors: the LC[TM1] connector of the optical link and the two flat cable connectors for the analogue differential input signals.

The pin assignment of the 34 pin input flat cable connectors and the termination circuitry is shown in figure 3. If one looks into the connector pin 1 is in the lower right corner. The upmost two pins 33 and 34 are connected to the analogue ground. The analogue differential input lines are terminated with 50 Ω and have an input range of −1.25 V to 0.25 V.

Furthermore there are ten LEDs. The two LEDs above the optical transceiver indicate the status of the optical link and should be always on during normal operation. The 'Clk' LED is lit when the fibre connection with the destination module was established successfully. The 'Syn' LED indicates that the clock recovery chip is active.

The two LEDs below the LC[TM] connector indicate the activity on the configuration bus that connects the HL FPGA with the ZR chips. The 'Com' LED indicates the activity on the data line. The 'Ack' LED shows the successful transmission of data packets.

Further below three LEDs display the loading status of the three FPGAs. The left LED is for the ZR0 chip, the middle one for ZR1 and the right one for the HL FPGA. During normal operation all three LEDs should be lit.

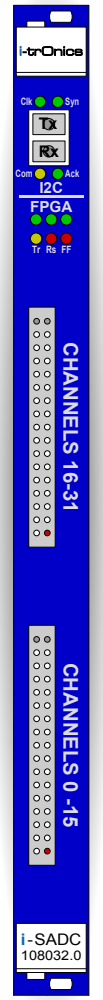

Figure 2: Front-panel

Figure 3: Analogue input connector pin-out

The last row of LEDs shows activity relevant for data taking. The left 'Tr' LED blinks when triggers are coming in. The middle 'Rs' LED shows the reset pulses and the 'FF' LED on the right side indicates overflows in the multi-hit buffer due to too high trigger and/or data rate. The 'FF' LED should be always off.

[1] LC[TM] is a trademark of Lucent

6.3 The VME backplane

The VME P1 and P2 connectors are only used for power connections. The **i-SADC108032** module needs approximately 25 W on the +5 V line and 10 W on the $-12V$ line. Make sure that the power supply of your VME crate is able to provide enough power for all inserted modules. For connector pin-out see tables 1 and 2.

A1	–	B1	–	C1	–
A2	–	B2	–	C2	–
A3	–	B3	–	C3	–
A4	–	B4	–	C4	–
A5	–	B5	–	C5	–
A6	–	B6	–	C6	–
A7	–	B7	–	C7	–
A8	–	B8	–	C8	–
A9	GND	B9	–	C9	GND
A10	–	B10	–	C10	–
A11	–	B11	–	C11	–
A12	–	B12	–	C12	–
A13	–	B13	–	C13	–
A14	–	B14	–	C14	–
A15	–	B15	–	C15	–
A16	–	B16	–	C16	–
A17	–	B17	–	C17	–
A18	–	B18	–	C18	–
A19	–	B19	–	C19	–
A20	–	B20	GND	C20	–
A21	–	B21	–	C21	–
A22	–	B22	–	C22	–
A23	–	B23	GND	C23	–
A24	–	B24	–	C24	–
A25	–	B25	–	C25	–
A26	–	B26	–	C26	–
A27	–	B27	–	C27	–
A28	–	B28	–	C28	–
A29	–	B29	–	C29	–
A30	–	B30	–	C30	–
A31	−12 V	B31	–	C31	+12 V
A32	+5 V	B32	+5 V	C32	+5 V

Table 1: VME backplane P1 connector pin-out

A1	reserved	B1	+5 V	C1	reserved
A2	reserved	B2	GND	C2	reserved
A3	–	B3	–	C3	reserved
A4	reserved	B4	–	C4	reserved
A5	reserved	B5	–	C5	reserved
A6	reserved	B6	–	C6	reserved
A7	reserved	B7	–	C7	reserved
A8	reserved	B8	–	C8	reserved
A9	reserved	B9	–	C9	reserved
A10	reserved	B10	–	C10	reserved
A11	reserved	B11	–	C11	reserved
A12	reserved	B12	GND	C12	reserved
A13	reserved	B13	+5 V	C13	reserved
A14	reserved	B14	–	C14	reserved
A15	reserved	B15	–	C15	reserved
A16	reserved	B16	–	C16	reserved
A17	reserved	B17	–	C17	reserved
A18	reserved	B18	–	C18	reserved
A19	reserved	B19	–	C19	reserved
A20	reserved	B20	–	C20	GND
A21	reserved	B21	–	C21	reserved
A22	reserved	B22	GND	C22	reserved
A23	GND	B23	–	C23	reserved
A24	reserved	B24	–	C24	GND
A25	GND	B25	–	C25	reserved
A26	reserved	B26	–	C26	GND
A27	GND	B27	–	C27	reserved
A28	reserved	B28	–	C28	GND
A29	GND	B29	–	C29	reserved
A30	reserved	B30	–	C30	reserved
A31	reserved	B31	GND	C31	reserved
A32	reserved	B32	+5 V	C32	reserved

Table 2: VME backplane P2 connector pin-out

6.4 Data processing and data flow

Figure 4 shows the flow of the ADC data on the board level. The data of ADC 0 to ADC 15 are going to the ZR0 FPGA, the data of ADC 16 to ADC 31 to the ZR1 FPGA. The ZR chips perform all the signal processing and data formatting and write their results via a 16 bit wide link to the HL FPGA. This chip multiplexes the data blocks from the two ZR FPGAs and sends them out to the HOTLink serializer from where they go to the optical transceiver.

All elements on the board are working synchronous to the clock signal which

is recovered from the incoming serial data stream of the optical transceiver.

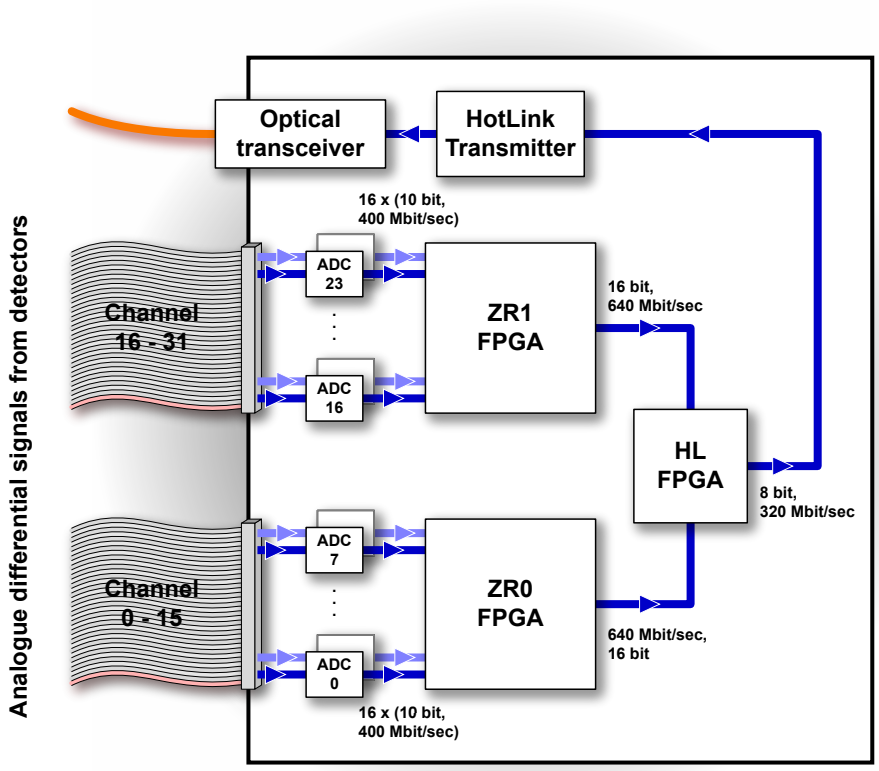

Figure 4: Flow of ADC data

Figure 5 shows the block diagram of the ZR FPGA. The 32 ADCs continuously digitise the incoming analogue signals. Each ZR FPGA buffers the sixteen 10 bit wide data streams in a shift register with user controllable depth. This allows to delay the ADC data and thus to compensate the trigger latency.

When the trigger pulse arrives a configurable number of samples is written into the multi-hit buffer, which can store the data of maximum four events. The number of samples per event is always constant and two events cannot overlap. If two trigger pulses are closer than the sampling window the ZR chips memorise that there was a trigger pulse and start writing the data of the second event right after the first one was finished. This means that in the sparse mode the minimum distance between two triggers is defined by the number of samples taken per event, because for shorter distances the time correlation of the signal in respect to the trigger in the data is lost.

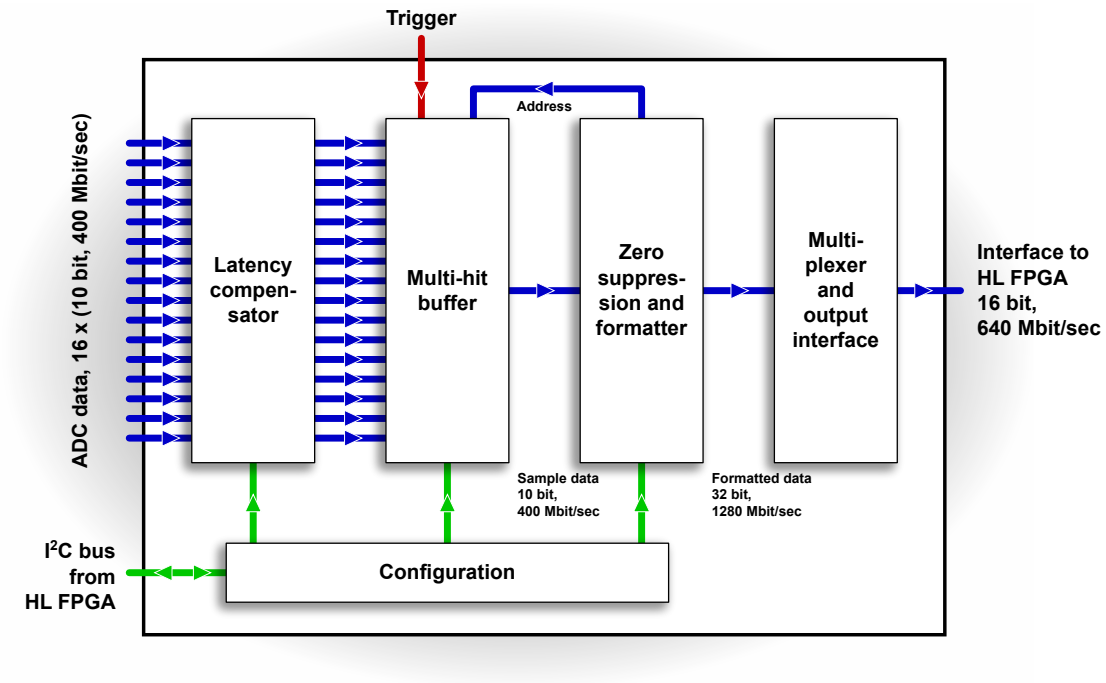
Trigger
Address
ADC data, 16 x (10 bit, 400 Mbit/sec)
Latency compen-sator
Multi-hit buffer
Zero suppres-sion and formatter
Multi-plexer and output interface
Interface to HL FPGA 16 bit, 640 Mbit/sec
Sample data 10 bit, 400 Mbit/sec
Formatted data 32 bit, 1280 Mbit/sec
I²C bus from HL FPGA
Configuration

Figure 5: Data processing in the ZR FPGA

The multi-hit buffer is read by the data processor unit. This unit computes for every event three user configurable sums over the array of samples: the base line integral (integral 0), the signal integral (integral 1) and the integral of the rising part of the signal (integral 2). For each integral the start index and the number of samples that are summed up can be configured independently.

To suppress channels that were not hit the data processor unit compares the difference of the signal integral (integral 1) and the base line integral (integral 0) with a threshold value that can be set for each channel individually. If the difference is below threshold the three integrals of the particular channel are not written out in sparse mode.

There is only on data processor unit which processes the channels one after the other so that the time needed for an event is a little bit more than sixteen times the sampling time. This time period defines the maximum average trigger rate in sparse mode. In latch-all mode this value is defined by the bandwidth of the optical link.

Consult section 8 on how to set the configuration parameters for latency compensation, number of samples, parameters of sample integrals and thresholds.

7 Data format

The data of the i-SADC108032 module are aligned in 32 bit words. For every event the module generates a data block. This event data block itself consists of two ADC data blocks. Each ADC data block is generated by one ZR FPGA and contains the data of the 16 channels connected to this chip.

The ADC data block starts with the ADC header word. The format of the ADC data itself depends on the data mode the i-SADC108032 module was set to. If the module is in latch all mode it sends out for every channel first the ADC sample values followed by the three words for the sample integrals. In sparse mode the module only writes out the words of the three sample integrals for each hit channel.

The definition of the ADC header word is shown in table 3. The lower twelve bits contain the local event number from the counter in the ZR FPGA. This number is meant to check the synchronicity with other modules and the global event number. If the maximum local event number of $(2^{12} - 1)$ is reached the counter will wrap around and start counting up from 0. In this way the 'Event number' field in the header should behave like the lower 12 bits of the global event number.

The second twelve bit wide field is the ADC block size which is the number of 32 bit words in the ADC data block including header word, so that the minimum ADC block size is one (the ADC header is sent for every event).

The bit at position 24 is the mode flag 'M'. If this bit is set to '0' the data have latch-all format. If the bit reads '1' the data are sparsified (zero-suppressed).

At position 25 sits the overflow flag 'O'. This bit indicates whether there was an overflow of the multi-hit buffer in the ZR FPGA due to a too high incoming trigger rate. This is also indicated by the 'FF' LED on the front-panel (see subsection 6.2). If the 'O' bit is set data are probably not valid, because the ZR FPGA took the data with an unknown time offset with respect to the trigger.

The last field in the ADC header is the 4 bit wide ADC ID. This ID serves to distinguish the different ZR FPGAs (and thus the connected channels) that are multiplexed by the i-MUX module. The i-MUX module series can multiplex the data of up to eight i-SADC108032 modules. With each ADC module having two ZR FPGAs this gives maximum 16 different IDs from 0 to 15.

In latch-all mode the header is followed by the words of the ADC sample data of the channel 0. Table 4 shows the format of these words. Each ADC sample word consists of three consecutive ten bit wide ADC values in ascending order

31										20								10									0

ADC header

10	ID (4)	O	M	ADC block size (12)	Event number (12)

Table 3: Definition of the ADC header.

with the first sample in the low part. The unused samples in the last word of the channel (depending on the configured number of samples per event; see subsection 8.3) are always set to zero.

The three ADC sample integral words are generated for every channel in latch-all mode and for every channel above threshold in sparse mode respectively. Their definition is shown in table 4. The two low bytes contain the sample integral itself. If this sum reaches the maximum value it saturates, which means that integral values of `0xFFFF` are not valid, because the real value of the sum is unknown.

The suppression flag 'S' at position 23 of the ADC sample integral word is a debugging feature in the latch-all mode where it indicates whether the channel would have been suppressed in the sparse mode ($S = 1$). This allows to cross-check the zero suppression.

The last field of this word is the 4 bit wide channel number. It is needed for the sparse mode, because in this mode only the hit channels are written out.

31										20								10									0

ADC sample data (latch all mode only)

10	Sample 2 (10)	Sample 1 (10)	Sample 0 (10)

ADC sample integral

10	Channel (4)	00	S	0000000	Sample integral (16)

Table 4: Definition of the ADC data words.

7.1 Decoding the data

The data decoding of the **i-SADC108032** module is straight forward. There are two cases to distinguish: the latch-all and the sparse mode.

In sparse mode only ADC sample integral words are generated. For each hit channel there are three of these words. So one reads blocks of three words as long as the number of read words is smaller than the ADC block size.

In latch-all mode there are in addition the ADC sample words for each channel. They come before the ADC sample integral words. The number of sample words per channel is calculated (block size − 1) divided by the number of channels (= 16) and decreased by the number of sample integral words (= 3). So for each channel one loops over the number of ADC sample words and then reads the three sample integral words like in sparse mode.

The following very simple code snippet illustrates the above procedure to encode the data of one ZR FPGA.

```
...

// bit fields for decoding
struct adc_hd_bit_field {
  unsigned int ev_nr  : 12;  // local event number
  unsigned int blk_sz : 12;  // block size
  unsigned int mode   : 1;   // data mode: '1' sparse, '0' latch all
  unsigned int ovfl   : 1;   // overflow in multi-hit buffer
  unsigned int id     : 4;   // ID of ZR chip
  unsigned int n_u    : 2;   // not used; should be '10'
};
adc_hd_bit_field *adc_hd;
struct adc_data_bit_field {
  unsigned int val_0 : 10;  // sample value 0
  unsigned int val_1 : 10;  // sample value 1
  unsigned int val_2 : 10;  // sample value 2
  unsigned int n_u   : 2;   // not used; should be '10'
};
adc_data_bit_field *sample_dt;
struct adc_integral_bit_field {
  unsigned int val   : 16;  // sample integral value
  unsigned int n_u_1 : 7;   // not used; should be 0
  unsigned int suppr : 1;   // suppression flag
  unsigned int n_u_2 : 2;   // not used; should be 0
  unsigned int ch_nr : 4;   // channel number
  unsigned int n_u_3 : 2;   // not used; should be '10'
};
adc_integral_bit_field *integral_dt;

int i = 0;  // index for data word array
int adc_cnt_wd, adc_cnt_sample_wd;  // word counters

...

// decode ADC header
adc_hd = (adc_hd_bit_field *) &data[i]; i++;

adc_cnt_wd = 1; // header is already read
```

```
// loop over words in ADC data block
while (adc_cnt_wd < adc_hd->blk_sz) {

  if (adc_hd->mode == 0) {
    adc_cnt_sample_wd = 0;
    // process latch all data
    while (adc_cnt_sample_wd < (adc_hd->blk_sz - 1) / 16 - 3) {
      // decode ADC sample data
      sample_dt = (adc_data_bit_field *) &data[i]; i++;
      adc_cnt_sample_wd++;
      adc_cnt_wd++;
    }
  }

  // process sample integrals
  for (int j = 0; j < 3; j++) {
    // decode ADC sample integrals
    integral_dt = (adc_integral_bit_field *) &data[i]; i++;
    adc_cnt_wd++;
  }
}

...
```

8 Configuration and loading

The FPGAs on the board store their firmware in volatile memories so they loose their programming whenever the power is interrupted. This means the first step of configuring the **i-SADC108032** module is loading the FPGAs.

8.1 Loading the HL FPGA

The firmware of the HL chip is stored in a Xilinx XC18V02VQ44 EEPROM. The PROM usually contains a valid firmware so that the HL FPGA is automatically programmed at each power up. The successful loading of the FPGA is indicated by the rightmost FPGA LED on the front-panel (see subsection 6.2).

If the LED is not lit or the HL firmware has to be updated the content of the PROM has to be rewritten. The PROM has a JTAG interface and to write a firmware file into it a Xilinx Parallel Cable III (or compatible programmer) and the iMPACT software is needed. The Xilinx Parallel Cable III has to be connected to the parallel port of the PC and the Molex plug to the corresponding jack on the **i-SADC** board (see figure 1). In order to be able to load the PROM the **i-SADC** module has to be powered at least via the +5 V line.

The iMPACT software is part of the ISE WebPACK suite that is available for free for Microsoft Windows 2000 and Windows XP at the Xilinx web site:

www.xilinx.com/xlnx/xil_prodcat_landingpage.jsp?title=ISE+WebPack

The ISE WebPACK suite comes in four flavours. For the PROM programming you need to download and install either the 'Complete' set or the 'Complete Programming Tools'.

To write the PROM file the following procedure can be used:

1. Start the iMPACT software.
2. In the 'Operation Mode Selection' panel choose 'Configure Devices' (default).
3. In the 'Configure Devices' panel choose 'Boundary-Scan Mode' (default).
4. In the 'Boundary-Scan Mode Selection' panel choose 'Automatically connect to cable and identify Boundary-Scan chain' (default).
5. Point to the MCS file you want to load.

6. When asked to select the device part name choose 'xc18v02_vq44'.
7. Click the device to select it.
8. Choose 'Program...' from the 'Operations' Menu.
9. Check 'Erase Before Programming', 'Verify' and 'Load FPGA'. Uncheck all other boxes.
10. Wait until the successful end of the operation is announced.
11. Make sure that the rightmost FPGA LED on the front-panel is lit indicating that the FPGA is programmed.

8.2 Loading the ZR FPGAs

The ZR FPGAs are loaded via the optical link and since the HL chip provides the necessary interface this chip has to be loaded beforehand (see subsection 8.1). In normal operation the firmware is transferred by the **i-MUX** module via the VME bus and the optical link of the **i-SADC108032** module. For more detailed description of the **i-MUX** VME interface consult the **i-MUX** manual. The successful programming of the ZR FPGAs is indicated by two LEDs on the front-panel (see subsection 6.2).

8.3 Configuring the ZR FPGAs

Only when the firmware of all three FPGAs is loaded the configuration registers in the ZR chips can be accessed. Table 5 gives an overview of the different registers and their addresses.

The latency register defines how many clock cycles the incoming ADC data are delayed. When set to the minimum value 0 the data are not delayed. The maximum value is 511 which corresponds to a delay of 12.78 μsec.

The second register sets the number of samples that are read per event. The maximum allowed value is 128.

The register at address 0x1002 is read only and holds the firmware version number of the ZR FPGA.

Registers 0x0004 to 0x0009 define the parameters for the three sample integrals that are computed for every event (see subsection 6.4). For each integral one can set the start index of the summation and the number of samples that contribute to the sum. Sample number 0 is the first sample in the event. The last accessible sample is at index (number of samples per event $-$ 1). This

Address (13 bits wide)	**Configuration value**	**Default value** (dec)	**Width**
0x0000	Latency in number of clock cycles	1	9
0x0001	Number of samples per event	128	9
0x0002	Firmware version (read only)	–	8
0x0003	Mode of data writing: 0 = latch-all, 1 = sparse	0	1
0x0004	Start index of sample integral 0	0	9
0x0005	Number of samples to sum up for integral 0	16	9
0x0006	Start index of sample integral 1	0	9
0x0007	Number of samples to sum up for integral 1	16	9
0x0008	Start index of sample integral 2	0	9
0x0009	Number of samples to sum up for integral 2	16	9
0x0010	Threshold 0	0	16
0x0011	Threshold 1	0	16
0x0012	Threshold 2	0	16
0x0013	Threshold 3	0	16
0x0014	Threshold 4	0	16
0x0015	Threshold 5	0	16
0x0016	Threshold 6	0	16
0x0017	Threshold 7	0	16
0x0018	Threshold 8	0	16
0x0019	Threshold 9	0	16
0x001A	Threshold 10	0	16
0x001B	Threshold 11	0	16
0x001C	Threshold 12	0	16
0x001D	Threshold 13	0	16
0x001E	Threshold 14	0	16
0x001F	Threshold 15	0	16
0x107F	Test register	0	16

Table 5: Address mapping of the internal configuration registers of the ZR FPGA

means the start index should always be smaller then the number of samples per event and the number of samples in the sum should be set to an appropriate value so that the integral does not leave the index range. Integral 0 is the baseline integral, integral 1 the signal integral and integral 2 can be assigned freely by the user, for instance as a integral of the rising part of the signal.

The threshold registers define the threshold values of the sixteen channels for the zero suppression. If the difference of the signal integral (integral 1) and the baseline integral (integral 0) is below the threshold value the channel is suppressed in sparse mode. For the ZR0 FPGA the threshold number corresponds to the input channel number, for the ZR1 chip the channel number

has an offset of 16.

The last register is just meant to be used to test the communication between the i-**SADC108032** and the connected i-**MUX**.